CYCLE 1

CYCLE 1

Date : 17-10-22
EXPERIMENT: 1.1

POLYNOMIAL ADDITION

AIM

Program to read two polynomials and store them in an array. Calculate the sum of the two polynomials and display the first polynomial, second polynomial and the resultant polynomial.

ALGORITHM

PROGRAM

```
//POLYNOMIAL SUM
#include <stdio.h>
#include<stdlib.h>

  typedef struct {
        int coeff;
        int expo;
             }
         polynomial;
 polynomial terms[100];
 int avail=0;

 //attach function
void attach(float coefficient,int exponent)
      {
      if(avail>=100)
      {
      printf("ERROR");
      exit(EXIT_FAILURE);
      }
      terms[avail].coeff=coefficient;
      terms[avail].expo=exponent;
      avail++;
      }
      //print function

      void print(int start,int finish)
      {    int i;
      for(i=start;i<finish;i++)
      {
      printf("%d x^%d + ",terms[i].coeff,terms[i].expo);
      }
      printf("%d x^%d ",terms[finish].coeff,terms[finish].expo);
      printf("\n");
      }

      //compare function
      int  COMPARE(int a ,int b)
        {
        if(a>b)
        return 1;
        else if (a==b)
        return 0;
        else if (a<b)
```

```
    return -1;
    }
  //polyadd function

    void padd( int startA,int finishA,int startB,int finishB,int startD,int finishD )
     {
      float coefficient;
      while(startA<=finishA && startB<=finishB)
      {
      switch(COMPARE(terms[startA].expo,terms[startB].expo))
      {
      case -1:
      attach(terms[startB].coeff,terms[startB].expo);
      startB++;
      break;

      case 0:
      coefficient=terms[startA].coeff+terms[startB].coeff;
      if(coefficient!=0)
      attach(coefficient,terms[startA].expo);
      startA++;
      startB++;
      break;

      case 1:
      attach(terms[startA].coeff,terms[startA].expo);
      startA++;
      break;
      }
      }

     for(;startA<=finishA;startA++)
     attach(terms[startA].coeff,terms[startA].expo);
     for(;startB<=finishB;startB++)
     attach(terms[startB].coeff,terms[startB].expo);
   }

//main function

  int main()
   {
   int size1,size2;
   int startA,startB,startD,finishA,finishB,finishD;
   printf("Enter the number of terms in the first polynomial and second polynomial\n");
   scanf("%d",&size1);
   scanf("%d",&size2);

   startA=0;
```

```
    finishA=size1-1;
    startB=size1;
    finishB=size2+size1-1;
    avail=size1+size2;



   printf("Enter the coefficent and exponent of polynomial 1 in the decreasing order of exponents");
   for(int i=startA;i<=finishA;i++)
{
scanf("%d%d",&terms[i].coeff,&terms[i].expo);
}
   printf("\nPolynomial 1 is \n ---------- \n");
   print(startA,finishA);
   printf("\n");
   printf("Enter the coefficent and exponent of polynomial 2 in the decreasing order of exponents");
   for(int i=startB;i<=finishB;i++)
   {
   scanf("%d%d",&terms[i].coeff,&terms[i].expo);
   }

  printf("Polynomial 2 is \n ---------- \n");
  print(startB,finishB);
  startD=avail;
  padd(startA,finishA,startB,finishB,startD,finishD);
  finishD=avail-1;
  printf("\n\nPOLYNOMIAL SUM\n----------\n");
  print(startD,finishD);
  printf("\n");
  return 1;
}
```

OUTPUT

```
Enter the number of terms in the first polynomial and second polynomial
3
2
Enter the coefficent and exponent of polynomial 1 in the decreasing order of exponents
3
2
2
1
1
0

Polynomial 1 is
```

```
 -------------------
3 x^2 + 2 x^1 + 1 x^0

Enter the coefficent and exponent of polynomial 2 in the decreasing order of exponents
2
1
1
0

Polynomial 2 is
 -------------------
2 x^1 + 1 x^0


POLYNOMIAL SUM
--------------
3 x^2 + 4 x^1 + 2 x^0
```

CYCLE 2

Date : 17-10-22
EXPERIMENT: 1.2

POLYNOMIAL EVALUATION

AIM

Program to read a polynomial of degree ‘n’ and store it in an array. Evaluate the polynomial for a given value of ‘x’.

ALGORITHM

PROGRAM

```
//program to evaluate a polynomail stored as array
#include <stdio.h>
#include <math.h>

typedef struct
{
   float coeff;
   int expo;
} poly;

int main()
{
   int len, x, result = 0;
   printf("Enter the no of terms  of the polynomial: ");
   scanf("%d", &len);
   poly a[len];
  printf("Enter the coefficents and exponent  of polynomial in descending order of exponents\n");
  for(int i=0;i<len;i++){
  scanf("%f %d",&a[i].coeff,&a[i].expo);
   }
  printf("Polynomial  is :\n");
  printf("----------- :\n");
  for (int i = 0; i < len-1 ;i++)
  {
  int x=(int)a[i].coeff;
  printf("%dx^%d +", x, a[i].expo);
  }
  for (int i = len-1; i < len; i++)
  {
   int x=(int)a[len-1].coeff;
   printf("%dx^%d", x, a[len-1].expo);
   }
   printf("\n");
   printf("Enter x\n");
   scanf("%d", &x);
   for (int i = 0; i <len; i++)
   result += a[i].coeff * pow(x, a[i].expo);
   printf("\nResult \n");
   printf("--------------------\n");
   printf(" %d\n", result);
   return 0;
}
```

OUTPUT

```
Enter the no of terms  of the polynomial: 5
Enter the coefficents and exponent  of polynomial in descending order of exponents
5
6
4
5
3
4
2
3
1
2
Polynomial  is :
---------- :
5x^6 +4x^5 +3x^4 +2x^3 +1x^2
Enter x
2

Result
-------------------
 516
```

CYCLE 1

Date : 17-10-22
EXPERIMENT: 1.3

SPARSITY EVALUATION

AIM

Program to represent and store a sparse matrix in an efficient way and find its sparsity.

ALGORITHM

PROGRAM

```
//sparse matrix representation of a matrix

 #include <stdio.h>
 int main()
 {
 int r,c;int n;
 typedef struct
  {
  int row;
  int col;
  int value;
 }element;
element a[100];
 printf("Enter the number of rows and columns of the matrix and non zero elements of the matrix  ");
 scanf("%d",&r);
 scanf("%d",&c);
 scanf("%d",&n);
 printf("Enter the row column and the value");

 for(int i=1;i<=n;i++)
 {

 scanf("%d%d%d",&a[i].row,&a[i].col,&a[i].value);
 }
 a[0].row=r;
a[0].col=c;
a[0].value=n;
 printf("\n Sparse matrix \n");
 for(int i=0;i<=n;i++ )
 {
 printf("%d %d %d ", a[i].row,a[i].col,a[i].value);
 printf("\n");
 }
printf("\n SPARSITY \n----------\n");
int zeroes=r*c-n;
float sparsity=(float)zeroes/(r*c);
printf("%f",sparsity);

  }
```

OUTPUT

Enter the number of rows and columns of the matrix and non zero elements of the matrix
3
3
4
Enter the row column and the value0
0
1
0
1
2
0
3
3
0
4
4

Sparse matrix
3 3 4
0 0 1
0 1 2
0 3 3
0 4 4

SPARSITY

0.555556

CYCLE 1

Date : 17-10-22
EXPERIMENT: 1.4

SPARSE MATRIX SUM

AIM

Program to input the representation of two sparse matrices and find the representation of their sum.

ALGORITHM

PROGRAM

```
#include<stdio.h>
int main(){
typedef struct{
int row;
int col;
int value;
}term;
term a[100],b[100],res[100];
int arow,brow,acol,bcol,anum,bnum;
printf("Enter the  row  ,column and non zero elements in sparse matrix 1 ");
scanf("%d%d%d",&arow,&acol,&anum);
printf("Enter the row ,column and non zero elements in sparse matrix 2 ");
scanf("%d%d%d",&brow,&bcol,&bnum);
//sparse matrix 1 input
printf("Enter the elements of sparse matrix 1 ");
for(int i=1;i<=anum;i++)
{
  scanf("%d%d%d",&a[i].row,&a[i].col,&a[i].value);
}
a[0].row=arow;
a[0].col=acol;
a[0].value=anum;

//sparse matrix 2
printf("Enter the elements of sparse matrix 2");
for(int i=1;i<=bnum;i++)
{
  scanf("%d%d%d",&b[i].row,&b[i].col,&b[i].value);
}
b[0].row=brow;
b[0].col=bcol;
b[0].value=bnum;


 //print matrix 1
 printf("SPARSE MATRIX 1\n");
 printf("----------------------\n");
 for(int i=0;i<=anum;i++)
 {
  printf("%d %d %d",a[i].row,a[i].col,a[i].value);
  printf("\n");
 }
```

```
//print matrix 2
 printf("SPARSE MATRIX 2\n");
printf("----------------------\n");
for(int i=0;i<=bnum;i++)
{
  printf("%d %d %d",b[i].row,b[i].col,b[i].value);
  printf("\n");
}


// to calculate sum

res[0].row=a[0].row;
res[0].col=a[0].col;

int p=1,q=1,r=1;
for(int i=0;i<b[0].row;i++)
{
for(int j=0;j<b[0].col;j++)
{
if(a[p].row==i && a[p].col==j && b[q].row==i && b[q].col==j)
    {
    res[r].row=i;
    res[r].col=j;
    res[r].value=a[p].value+b[q].value;
    r++;
    p++;
    q++;

    }
else if(a[p].row ==i&&a[p].col==j)
{

res[r].row=i;
res[r].col=j;
res[r].value=a[p].value;
r++;
p++;
}
else if(b[q].row==i&&b[q].col==j)
{
 res[r].row=i;
 res[r].col=j;
 res[r].value=b[q].value;
 q++;
 r++;
}
```

```
 }
 }
res[0].value=--r;
printf(" SUM SPARSE : \n—----------\n");
for(int i=0;i<=res[0].value;i++)
{
       printf("%d\t%d\t%d\n",res[i].row,res[i].col,res[i].value);

   }

    return 0;

 }
```

OUTPUT

```
Enter the  row  ,column and non zero elements in sparse matrix 1 3
4
3
Enter the row ,column and non zero elements in sparse matrix 2 3
4
4
Enter the elements of sparse matrix 1 0
0
9
1
1
-6
2
0
11
Enter the elements of sparse matrix 20
1
7
1
1
-4
1
2
8
2
1
```

```
6
SPARSE MATRIX 1
----------------------
3 4 3
0 0 9
1 1 -6
2 0 11
SPARSE MATRIX 2
----------------------
3 4 4
0 1 7
1 1 -4
1 2 8
2 1 6
SUM SPARSE
—----------------
3       4       6
0       0       9
0       1       7
1       1       -10
1       2       8
2       0       11
2       1       6
```

CYCLE 1

Date : 17-10-22
EXPERIMENT: 1.5

FAST TRANSPOSE

AIM

Program to input the representation of a sparse matrix and find the representation of its transpose.

ALGORITHM

PROGRAM

```
#include <stdio.h>
#define max_terms 101
typedef struct
{
   int row;
   int col;
   int value;
} matrix;

void fast_transpose(matrix a[], matrix b[], int size)
{
   int row_terms[max_terms];
   int starting_pos[max_terms];
   int num_cols = a[0].col, num_terms = a[0].value;
   b[0].row = num_cols;
   b[0].col = a[0].row;
   b[0].value = num_terms;

   if (num_terms > 0)
   {
      for (int i = 0; i < num_cols; i++)
         row_terms[i] = 0;
      for (int i = 1; i <= num_terms; i++)
         row_terms[a[i].col]++;
      starting_pos[0] = 1;
      for (int i = 1; i < num_cols; i++)
         starting_pos[i] = starting_pos[i - 1] + row_terms[i - 1];
      for (int i = 1; i <= num_terms; i++)
      {
         int j = starting_pos[a[i].col]++;
         b[j].row = a[i].col;
         b[j].col = a[i].row;
         b[j].value = a[i].value;
      }
   }
   printf("Transpose of sparse matrix representation\n");
   for (int i = 0; i <= size; i++)
   {
      printf("%d %d %d\n", b[i].row, b[i].col, b[i].value);
   }
}
```

```
int main()
{
   int size;
   printf("Enter the number of non zero elements: ");
   scanf("%d", &size);

   matrix m[30], t_m[30];
   m[0].value = size;

   printf("Enter the number of rows and columns: ");
   scanf("%d %d", &m[0].row, &m[0].col);

   printf("Enter the sparse matrix representation: ");
   for (int i = 1; i <= size; i++)
   {
      scanf("%d %d %d", &m[i].row, &m[i].col, &m[i].value);
   }

   printf("Sparse matrix representation\n");
   for (int i = 0; i <= size; i++)
   {
      printf("%d %d %d\n", m[i].row, m[i].col, m[i].value);
   }

   fast_transpose(m, t_m, size);
}
```

OUTPUT

```
Enter the number of non zero elements: 4
Enter the number of rows and columns:
4
5
Enter the sparse matrix representation: 0
0
9
0
3
17
2
1
```

-16
3
1
10
Sparse matrix representation
4 5 4
0 0 9
0 3 17
2 1 -16
3 1 10
Transpose of sparse matrix representation
5 4 4
0 0 9
1 2 -16
1 3 10
3 0 17